

Date⇒ 16-02-2021

Module⇒ Backend

Lecture By⇒ Akash Handa

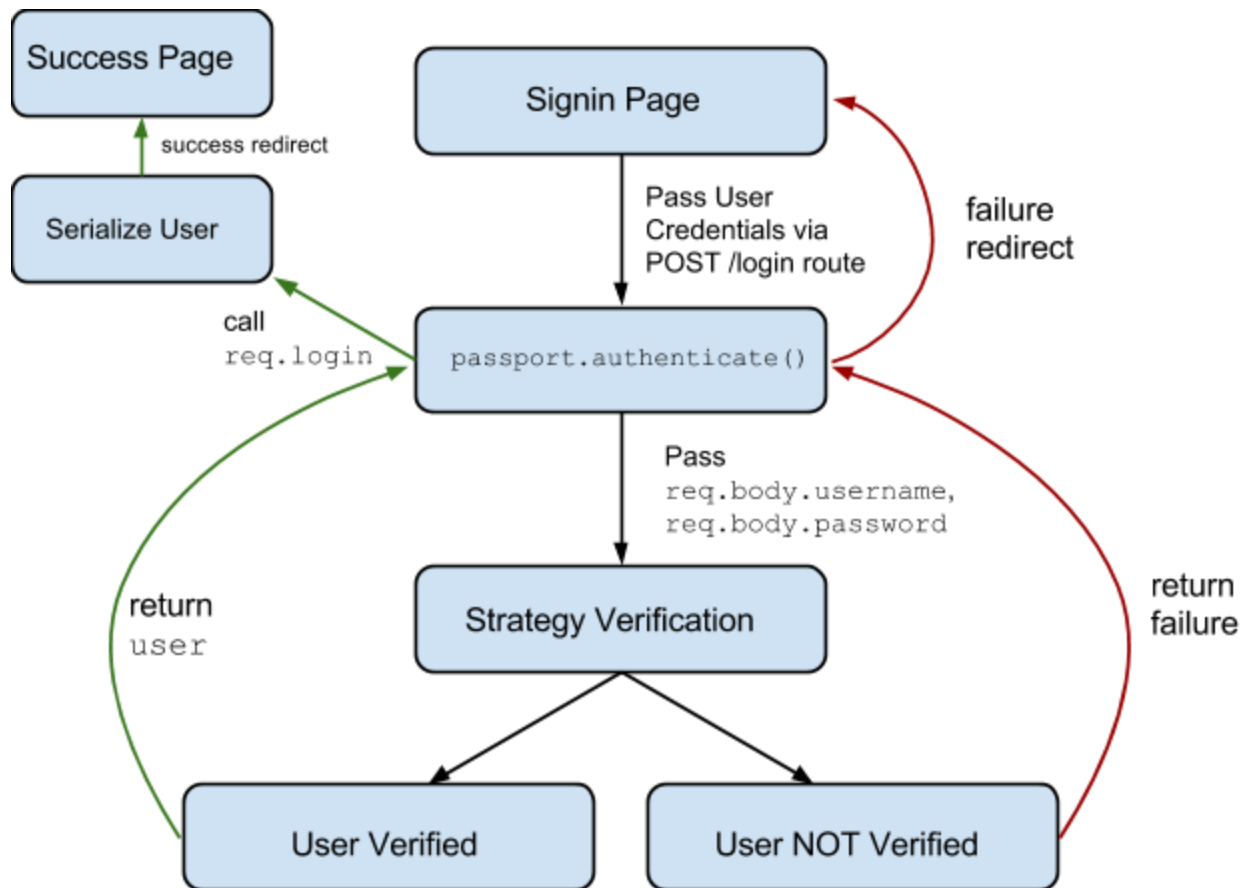
Subject ⇒ Passport Js

IN PREVIOUS LECTURE (QUICK RECAP) Date-15/02/2021	In Today's Lecture (Overview)
What is JWT Packages Required For LoginwithJWT What is Mongoose?	

What is Passport js

Passport is authentication middleware for Node. **js**. Extremely flexible and modular, **Passport** can be unobtrusively dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Facebook, Twitter, and more. 500+ Strategies Now!

Passport.js out of the box is safe as your implementation of it to **protect** routes from unauthorized access. For example if you forget to apply the middleware to certain routes they would not be protected, if you make a mistake in configuring the authentication strategy you may open up your application to an attack.



What is SuperAgent

Small progressive client-side HTTP request library, and Node.js module with the same API, supporting many high-level HTTP client features

<https://www.npmjs.com/package/superagent>

Login with Google

App.js

```
const express = require('express');
const app = express();
const session = require('express-session');
const passport = require('passport');
const port = 9800;
```

```
const GoogleStrategy = require('passport-google-oauth').OAuth2Strategy;

//start passport
app.use(passport.initialize());
app.use(passport.session());
app.set('view engine', 'ejs');

passport.serializeUser((user, cb) => {
  cb(null, user)
})

app.get('/', (req, res) => {
  res.render('pages/login')
});

app.get('/profile', (req, res) => {
  res.send(userprofile)
})

app.get('/error', (req, res) => res.send('Error while login'));

passport.use(new GoogleStrategy({
  clientID: '',
  clientSecret: '',
  callbackURL: "http://localhost:9800/auth/google/callback"
},
function(accessToken, refreshToken, profile, done) {
  userprofile = profile;
  return done(null, userprofile)
}
));

//will take to google website
app.get('/auth/google',
  passport.authenticate('google', { scope:
['https://www.googleapis.com/auth/plus.login']
})));

//when thing done on google it come back
```

```

app.get('/auth/google/callback',
  passport.authenticate('google', { failureRedirect: '/error' }),
  function(req, res) {
    res.redirect('/profile');
  });

app.listen(port, () => {
  console.log(`App is running on port ${port}`)
})

```

Login.ejs

```

<a href="/auth/google">Sign In with Google</a>

```

Login with github

App.js

```

const express = require('express');
const app = express();
const superagent = require('superagent');
const request = require('request');
const port = 7800;

app.set('view engine', 'ejs');

app.get('/', (req, res) => {
  res.render('pages/login.ejs')
})

app.get('/profile', (req, res) => {
  const code = req.query.code;
  if(!code) {
    res.send({
      success: false,
      message: 'Error on access'
    })
  }
  superagent

```

```

    .post('https://github.com/login/oauth/access_token')
    .send({
      client_id: '100cc18d754140bbced1' ,
      client_secret: '7d4ae1d52ca894890ee0afc17309c63cc1a0b23d',
      code: code
    })
    .set('Accept', 'application/json')
    .end((err, result) => {
      if (err) throw err;
      var accesstoken = result.body.access_token;
      var option = {
        url: 'https://api.github.com/user',
        method: 'GET',
        headers: {
          'Accept': 'application/json',
          'Authorization': 'token ' + accesstoken,
          'User-Agent': 'mycode'
        }
      }
      request(option, (err, response, body) => {
        return res.send(body)
      })
    })
  })
})

app.listen(port, () => {
  console.log(`Server is running on port ${port}`)
})

```

Login.ejs

```

<a
href="https://github.com/login/oauth/authorize?client_id=100cc18d754140bbced1">
  Login With Github
</a>

```